DBIF Tag & Channel Mechanism

Integrated Test & Operations System
Developer Documentation
5 October 2006

Proc Widget

The Proc widget is the widget STOL uses to display executing procs.

Synopsis

```
#include <Proc.h>

widget = XtVaCreateManagedWidget("widget-name",
                        procWidgetClass, parent,
                        va-arglist);
```

Description

The Proc widget is a scrolled window that is used to view lines in a file and optionally highlight one line (the 'current' line) in that file.

The Proc widget was originally developed to display STOL procs.

The Proc widget operates in either of two modes. In autoscroll mode, the vertical scrollbar is insensitive and the view automatically scrolls so the current line remains visible in the view. In scrollbar mode, the vertical scrollbar is sensitive and controls scrolling.

Resources

NAME	CLASS	TYPE	DEFAULT VALUE	ACCESS
autoscroll	Autoscroll	Boolean	TRUE	CSG
clinenobg	Foreground	Pixel	XtDefaultForeground	C G
clinenofg	Background	Pixel	XtDefaultBackground	C G
curlinebg	Foreground	Pixel	XtDefaultForeground	C G
curlinefg	Background	Pixel	XtDefaultBackground	C G
curlineno	Curlineno	int	0	CSG
font	Font	XFontStruct *	XtDefaultFont	C G
hbar	Scrollbar	Widget	dynamic	G
hideHoriz	HideScrollbar	Boolean	FALSE	CSG
leftMargin	Margin	int	4	C G
lineMargin	Margin	int	0	C G
linenobg	Background	Pixel	XtDefaultBackground	C G
linenofg	Foreground	Pixel	XtDefaultForeground	C G
linenoFont	Font	XFontStruct *	XtDefaultFont	C G
linenoMargin	Margin	int	0	C G
linenoon	Linenoon	Boolean	TRUE	C G
maxwidth	Maxwidth	int	dynamic	G
nlines	Nlines	int	dynamic	G
normalbg	Background	Pixel	XtDefaultBackground	C G
normalfg	Foreground	Pixel	XtDefaultForeground	C G
procfile	Procfile	StringArray	NULL	C G
topMargin	Margin	int	4	C G
vbar	Scrollbar	Widget	dynamic	G

view	DrawingArea	Widget	dynamic	G
viewHeight	ViewHeight	int	1	C G
viewWidth	ViewWidth	int	1	C G

Access: C – resource may be set at creation time; G – resource may be retrieved using XtVaGetValues; and S – resource may be set using XtVaSetValues.

autoscroll Controls autoscroll mode vs. scrollbar mode.

When *autoscroll* is TRUE, the Proc widget is in autoscroll mode. The vertical scrollbar is insensitive and scrolling is automatic, so that, as the proc executes, the current line remains visible in the view. Setting *autoscroll* TRUE scrolls so the current line is visible.

When *autoscroll* is FALSE, the Proc widget is in scrollbar mode. The vertical scrollbar is sensitive and controls scrolling.

clinenobg
clinenofg
curlinebg
curlinefg
linenobg
linenofg
normalbg
normalfg Foreground and background colors used to draw text and linenumbers. *curlinefg* and *curlinebg* are used to draw the text of the current line; *clinenofg* and *clinenobg* are used to draw the current line's line number. *normalfg* and *normalbg* are used to draw the text of all other lines; *linenofg* and *linenobg* are used to draw the line number of all other lines.

By default, the current line is drawn in reverse video.

curlineno Line number of the current line (line number 1 is the first line in the proc file). If *curlineno* gets set to 0 (or any other number that doesn't correspond to a line in the proc file) there is no current line.

font
linenoFont The fonts used to draw lines and line numbers. *font* is used to draw the text of lines and *linenoFont* is used to draw line numbers.

hbar
vbar
view The widget ID of the child widgets. *hbar* and *vbar* are the horizontal and vertical scrollbar widgets (class *XmScrollBar*); *view* is the view widget (class *XmDrawingArea*).

hideHoriz Controls whether or not the horizontal scrollbar is visible. When *hideHoriz* is *TRUE*, the horizontal scrollbar is not displayed no matter how narrow the view. Otherwise the horizontal scrollbar will be displayed whenever the view is too narrow to display the longest line in the proc.

hideHoriz is *TRUE* by default.

There is no corresponding resource to control display of the vertical scrollbar. The vertical scrollbar is displayed whenever the view is too short to display all the lines in the proc.

leftMargin
lineMargin
linenoMargin
topMargin Various margins in pixels. *leftMargin* is the margin between the left edge of the view and the line number (or line text when *linenoon* is *False*. *lineMargin* is additional space between lines. *linenoMargin* is additional space between the line number and the line text. *topMargin* is the margin between the top edge of the view and text.

linenoon Flag indicating whether or not to draw line numbers; line numbers are drawn when *linenoon* is *TRUE*.

Line numbers are drawn by default.

maxwidth
nlines Number of lines in *procfile* and number of chars in the widest line. These are read-only resources that may not be set using XtVaSetValues().

procfile

The array of strings comprising the proc. Each string represents a line. The array and aach string in the array must be XtMalloc'd. The end of the array is marked by a NULL pointer. Strings should not contain `'\t'` or `'\n'` characters. The Proc widget will use this array directly; the application must not alter this array (or any of the strings) after creating the Proc widget. The following could be used as a *procfile*:

```
String procfile[];
procfile = (String *) XtMalloc(4 * sizeof(String));
procfile[0] = XtNewString("proc sayhi");
procfile[1] = XtNewString("shoval \"hi mom\"");
procfile[2] = XtNewString("endproc");
procfile[3] = NULL;
```

viewWidth
viewHeight

The initial size, in characters and lines, of the Procwidget's view.

Other commonly set resources include

**procwidget*`.borderWidth`
**procwidget*`.borderColor`
**procwidget*`.view.background`

Programmatic interface

`String` ProcGetCurlineno (`Widget` *w*) Function

`String *` **ProcRead** (`FILE *`*fp*) Function

A utility function to read a proc file.

```
fd = fopen(procfile,"r");
proc = XtVaCreateManagedWidget("proc",
                procWidgetClass, parent_widget,
                "procfile", ProcRead(fd),
                NULL, 0);
fclose(fd);
```

`void` **ProcRefresh** (`Widget` *w*, `XEvent *x`, `String *y`, `Cardinal *z`) Function

Redraws *w*. **ProcRefresh**() is actually an action routine; *x*, *y*, and *z* are unused. For convenience, *NULL* should be passed for *x*, *y*, and *z*.